JOURNAL OF COMPUTER AND SYSTEM SCIENCES 22, 144–177 (1981)

Decision Problems for Node Label Controlled Graph Grammars

D. JANSSENS

Department of Mathematics, University of Antwerp UIA, B-2610 Wilrijk, Belgium

AND

G. ROZENBERG

Institute of Applied Mathematics and Computer Science, University of Leiden, Leiden, The Netherlands

Received October 25, 1979

Two basic techniques are presented to show the decidability status of a number of problems concerning node label controlled graph grammars. Most of the problems are of graph-theoretic nature and concern topics like planarity, connectedness and bounded degreeness of graph languages.

INTRODUCTION

The theory of graph grammars constitutes a well-motivated and by now well-developed area within theoretical computer science, see, e.g., [2] and in particular [3, 4].

Node label controlled graph grammars (NLC grammars) were introduced in [5] and further investigated in [6]. They were introduced in an attempt to formulate sequential (as opposed to parallel) graph grammars in which both basic components of a single rewriting step ((1) the (node) replacement, and (2) the establishing of the connections between the "daughter" graph introduced by rewriting and the rest of the "mother" graph) are dependent on node labels only and not on the "node configuration." In this way NLC grammars model a process of rewriting of graphs which is quite "context independent," where it should be immediately stated that the phrase "context independent" in graphs has quite a different meaning than in strings (there is simply so much more structure to be dependent on). More specifically if a node v in a graph A is rewritten by an NLC grammar into a graph B then (1) the choice of B depends on the label of v only, and (2) the way that connections between a node u of B a node $\bar{v}$ of A (adjacent to v) are established is dependent on the labels of u and $\bar{v}$ only.

In [5] we have investigated the structure of derivations in NLC grammars and we have arrived at a "pumping theorem" for NLC grammars. This is quite surprising, especially so that in [6] it is shown that, quite contrary to the original motivation, NLC grammars still possess many "context dependencies." In [6] we also investigate

0022-0000/81/020144–34$02.00/0

various restrictions and extensions of NLC grammars, provide some normal forms for them and also investigate the connection between NLC grammars and string languages.

Now we turn to basic decision problems concerning NLC grammars. Although we briefly consider some standard language theoretic decision problems, such as the decision status of the equivalence problem of NLC grammars, the bulk of the paper is devoted to problems intrinsic to *graph* grammars and languages. Hence we consider problems concerning planarity, connectivity, bounded degreeness of graphs (in a given graph language), etc. It is worthwhile to stress at this point that the results we obtain not only shed light (we think) on the "effectiveness of graph language generation by graph grammars" but also, their proofs illustrate a number of examples of quite involved graph language defining mechanisms that NLC grammars provide.

The paper is organized as follows.

In Section I we recall basic notions and terminology concerning graphs and graph grammars, providing at the same time basic notation for this paper.

In Section II we solve "directly" two basic decision problems concerning NLC grammars. In this way the reader gets acquainted with NLC grammars.

In Section III we present our first basic technique for proving the undecidability of some problems concerning NLC grammars. It consists of a rather elaborate way of coding (an instance of) the Post Correspondence Problem into the language of a NLC grammar. Using this technique we show the undecidability of several problems of graph theoretical nature concerning NLC grammars.

In Section IV we show a different technique for proving the undecidability of a problem concerning NLC grammars. It consists of reducing the problem to the emptiness problem for context sensitive (string) grammars. We show several applications of this technique.

I. Preliminaries

In this section we recall some basic notions and terminology concerning graphs and graph grammars. We also establish the basic notation for this paper.

For a finite set X, $\#X$ denotes its cardinality. The set of non-negative integers is denoted by $\mathbb{N}$ and the set of positive integers is denoted by $\mathbb{N}^+$.

As far as (undirected labelled) graphs are concerned we use the following notation and terminology. (Notions that are not explained in this section can be found in [1]).

(1) In the sequel, unless otherwise indicated, we will consider undirected, labelled graphs, shortly called "*graphs.*" An (*undirected labelled*) graph is a 4-tuple (V, E, Σ, φ), where V is a finite nonempty set (of *nodes*), E is a set of multisets of two elements of V (the set of *edges*), Σ is a finite nonempty set (of *labels*), and φ is a function from V to Σ (the *labelling function*). If X is a graph, then we will denote the set of nodes, the set of edges, the set of labels and the labelling function of X by V_X, E_X, Σ_X, φ_X, respectively.

(2) If Σ is a finite set and X is a graph with $\Sigma_X \subseteq \Sigma$ then X is a *graph over* Σ. We denote the set of graphs over Σ by G_Σ.

(3) Two nodes of a graph are called neighbors (or *adjacent*) if they are incident to the same edge. A graph X is *totally disconnected* if $E_X = \varnothing$.

(4) An *isolated node* is a node that is not incident to any edge.

(5) The *degree of a node* in a graph is the number of edges that are incident to it.

(6) *The degree of a graph*, *deg*(H), is the maximal degree of its nodes.

(7) A set L of graphs is of *degree* k if for all $H \in L$ we have *deg*$(H) \leqslant k$ and there is a graph $H \in L$ such that *deg*$(H) = k$.

(8) A set L of graphs is of *bounded degree* if there exists a positive integer k such that L is of degree k.

(9) A *chain* is a graph of the form

$$\begin{array}{ccccccc} l^{(1)} & l^{(2)} & l^{(3)} & l^{(4)} & & l^{(n-1)} & l^{(n)} \\ \bullet\!-\!\!-\!\!-\!\!-\!\!-\!\! & \bullet\!-\!\!-\!\!-\!\!-\!\!-\!\! & \bullet\!-\!\!-\!\!-\!\!-\!\!-\!\! & \bullet\!-\!\!-\!\! & \cdots & -\!\!-\!\!\bullet\!-\!\!-\!\!-\!\! & \bullet \\ x_0 & x_1 & x_2 & x_3 & & x_{n-1} & x_n \end{array}$$

(two nodes are of degree 1 and all the other nodes are of degree 2). We will denote the labels above the chain and the names of the nodes under it.

(10) If B is a subgraph of a graph A, then $A \backslash B$ denotes the full subgraph of A with node set $V_A \backslash V_B$. If B has only one node, say b, then we write $A \backslash b$ instead of $A \backslash B$.

(11) If Σ, $\bar{\Sigma}$ are finite nonempty sets, then a *weak coding* h from Σ into $\bar{\Sigma}$ is a function from Σ to $\bar{\Sigma} \cup \{\Lambda\}$. The image under h of a graph (V, E, Σ, φ) is the graph $(\bar{V}, \bar{E}, \bar{\Sigma}, \bar{\varphi})$ over $\bar{\Sigma}$, where

$$\bar{V} = V \backslash \{x \in V \mid h(\varphi(x)) = \Lambda\},$$

$$\bar{E} = \{\{x, y\} \in E \mid x \in \bar{V}, y \in \bar{V}\},$$

$$\bar{\varphi} \text{ is } h \circ \varphi \text{ restricted to } \bar{V}.$$

If $h(\Sigma) \subseteq \bar{\Sigma}$ then we say that h is a *coding*.

(12) Two graphs X and Y are *isomorphic* if there is a bijection h from V_X to V_Y such that $\varphi_Y \circ h = \varphi_X$, and $\{a, b\} \in E_X$ if and only if $\{h(a), h(b)\} \in E_Y$.

Next we will recall from [5, 6] the basic notions, terminology and notation concerning node label controlled graph grammars.

(13) A NLC *grammar* is a system $G = (\Sigma, \Delta, P, C, Z)$, where
- Σ is a finite nonempty set (the *total alphabet*),
- Δ is a nonempty subset of Σ (the *terminal alphabet*),
- P is a finite set of pairs of the form (d, D), where $d \in \Sigma$ and $D \in G_\Delta$; P is called the *set of productions*,
- C is a subset of $\Sigma \times \Sigma$ (the *connection relation*),
- Z is a graph over Σ (the *axiom*).

To define a derivation step in a NLC grammar $G=(\Sigma, \Delta, P, C, Z)$ we first define a "concrete" derivation step: if H, $\bar{H}$ are graphs over Σ, $v \in V_H$ and $\varphi_H(v)=d$, then H *directly c-derives* $\bar{H}$ *by replacing node* v by $\bar{D}$, *using production* (d, D) (denoted $H \Rightarrow_{(v,D)} \bar{M})$ if there exists a graph $\bar{D}$, isomorphic to D with $V_H \cap V_{\bar{D}} = \phi$ and such that

(a) $\quad V_{\bar{H}} = (V_H \backslash \{v\}) \cup V_{\bar{D}}$,

(b) $\quad E_{\bar{H}} = (E_H \backslash \{\{x, y\} \mid x = v \text{ or } y = v\}) \cup E_{\bar{D}}$
$\quad\quad \cup \{\{x, y\} \mid x \in V_{\bar{D}}, y \in V_H\{v\}, (\varphi_{\bar{D}}(x), \varphi_H(y)) \in C\}$,

(c) $\quad \varphi_{\bar{H}}(x) = \begin{cases} \varphi_H(x) & \text{if } \ x \in V_H, \\ \varphi_D(x) & \text{if } \ x \in V_{\bar{D}}. \end{cases}$

Intuitively speaking this means that the d-labelled node v is removed from H omitting all the edges incident to v; then v is replaced by $\bar{D}$, which is isomorphic to D, and finally new edges are added according to the connection relation C: if x is a node of $\bar{D}$ and y is a neighbor of v in H, then a new edge, incident to x and y is added if and only if $(\varphi_D(x), \varphi_H(y)) \in C$. If H, $\bar{H}$ are graphs over Σ, then H *directly derives* $\bar{H}$ in G, denoted $H \Rightarrow_G \bar{H}$, if there exists a node $v \in V_H$, a production $(\varphi_H(v), D) \in P$ and a graph $\hat{H}$ over Σ such that $H \Rightarrow_{(v,D)} \hat{H}$ and $\hat{H}$ is isomorphic to $\bar{H}$.

For a non-negative integer n and graphs H, $\bar{H}$ over Σ we will write $H \Rightarrow_G^n \bar{H}$ if either $n=0$ and $H=\bar{H}$, or $n=1$ and $H \Rightarrow_G \bar{H}$, or there are graphs $H_1, H_2, \dots, H_{n-1}$ over Σ such that $H \Rightarrow_G H_1 \Rightarrow_G H_2 \Rightarrow_G \cdots \Rightarrow_G H_{n-1} \Rightarrow_G \bar{H}$.

We will denote the reflexive and transitive closure of $\Rightarrow_G$ by $\Rightarrow_G^*$ and the transitive closure by $\Rightarrow_G^+$.

If $G(\Sigma, \Delta, P, C, Z)$ is a NLC grammar, then the *exhaustive* language $S(G)$ of G is the set $\{H \mid Z \Rightarrow_G^* H\}$. The language *generated by* G, denoted $L(G)$ is the set $S(G) \cap G_\Delta$.

A *derivation* $\mathscr{D}$ of a graph H in a NLC grammar G is a sequence $H_0 = Z, H_1, H_2, \dots, H_n = H$ such that, for $0 \leqslant i \leqslant n-1$, $H_i \Rightarrow_G H_{i+1}$ together with a precise description of how the rewriting $H_i \Rightarrow_G H_{i+1}$ is performed. This is formally defined in [5]. The number of steps, n, is then called the *length of* $\mathscr{D}$, H is the *result of* $\mathscr{D}$ and $H_1, H_2, \dots, H_{n-1}$ are the *intermediate graphs* of $\mathscr{D}$.

In the sequel, we assume that a NLC grammar $G=(\Sigma, \Delta, P, C, Z)$ does not have useless symbols, that is, for all $d \in \Sigma$, there are graphs (over Σ) H, $\bar{H}$ such that $Z \Rightarrow_G^* H \Rightarrow_G^* \bar{H}$, H has at least one d-labelled node, and $\bar{H} \in L(G)$.

(14) For a NLC grammar $G=(\Sigma, \Delta, P, C, Z)$ we define $L_{\text{conn}}(G) = \{H \mid \in L(G)$ and H is conneted$\}$, $S_{\text{conn}}(G) = \{H \mid H \in S(G)$ and H is connected$\}$.

(15) In [6] we also introduce the concept of a RNLC grammar. This differs from a NLC grammar by the fact that in a RNLC grammar every production has its own connection relation. A production of a RNLC grammar $G=(\Sigma, \Delta, P, Z)$ is thus of

the form $\pi = ((d, D); C(\pi))$, where (d, D) is as in the case of a NLC grammar, and $C(\pi)$ is a subset of $\Sigma \times \Sigma$; the *connection relation* of π. The definition of

$$H \underset{(v,D)}{\Rightarrow} \bar{H} \text{ using the production } ((d, D); C(\pi))$$

is analogous to that for NLC grammars, given in (14), except that (b) becomes

$$E_{\bar{H}} = (E_H \backslash \{\{x, y\} \mid x = v \text{ or } y = v\}) \cup E_{\bar{D}}$$
$$\cup \{\{x, y\} \mid x \in V_{\bar{D}}, y \in V_H \backslash \{v\}, (\varphi_{\bar{D}}(x), \varphi_M(y)) \in C(\pi)\}.$$

(16) Two grammars G, $\bar{G}$ with $L(G) = L(\bar{G})$ are called *equivalent*. It is shown in [6] that there is an algorithm which, given an arbitrary RNLC grammar G constructs an equivalent NLC grammar $\bar{G}$. On the other hand it is easy to see that for any NLC grammar $\bar{G}$ one can construct an equivalent RNLC grammar G: associate the connection relation of $\bar{G}$ with every production of $\bar{G}$.

(17) Furthermore, it is shown in [6] that there is an algorithm that for an arbitrary NLC grammar G constructs an equivalent NLC grammar $\bar{G} = (\bar{\Sigma}, \bar{\Delta}, \bar{P}, \bar{C}, \bar{Z})$ such that $\bar{P}$ contains no productions for elements of $\bar{\Delta}$. From (16) it follows that an analogous algorithm exists for RNLC grammars.

In this paper we will also consider an extension of a NLC grammar resulting by allowing erasing productions.

DEFINITION. A NLC *grammar with erasing*, denoted ΛNLC *grammar*, is a system $G = (\Sigma, \Delta, P, C, Z)$, where $P = P_1 \cup P_2$, $(\Sigma, \Delta, P_1, C, Z)$ is an NLC grammar and P_2 is a set of productions of the form (d, Λ) with $d \in \Sigma$; the elements of P_2 are called *erasing productions*.

We let $H \Rightarrow_{(v,\Lambda)} \bar{H}$ by applying production (d, Λ) if $v \in V_H$, $\varphi_H(v) = d$, $V_{\bar{H}} = V_H \backslash \{v\}$, and $E_{\bar{H}} = (E_H \backslash \{\{v, x\} \mid x \in V_H\}) \cup \{\{x, y\} \mid x \neq y$ and $\{v, x\}$, $\{v, y\} \in E_H\}$. This means that the d-labelled node v is removed and that all neighbors of a in H become adjacent to each other in $\bar{H}$. The application of productions from P_1 is defined as for NLC grammars. ∎

We end this section by defining several additional notions concerning graphs and NLC graph grammars that will be used in the sequel.

DEFINITION. (1) Let $H = (V_H, E_H, \Sigma_H, \varphi_H)$ be a graph. Then

(1.1) $x, y \in E_H$ are *adjacent labels in H* if there exist n_1, $n_2 \in V_H$ such that $\varphi_H(n_1) = x$, $\varphi_H(n_2) = y$, and $\{n_1, n_2\} \in E_H$.

(1.2) $X, Y \subseteq \Sigma_H$ are *adjacent sets of labels in H* if there exist $x \in X$, $y \in Y$ such that x and y are adjacent labels in H.

(1.3) $x, y \in \Sigma_H$ are *connected labels in H* if there exist $n_1, n_2 \in V_H$ such that $\varphi_H(n_1) = x$, $\varphi_H(n_2) = y$ and there exist nodes $v_1, v_2, \ldots, v_i$ in V_H such that $v_1 = n_1$, $v_i = n_2$ and for $j \in \{1, 2, \ldots, i-1\}$, v_j is adjacent to v_{j+1}.

(1.4) $X, Y \subseteq \Sigma_H$ are *connected sets of labels in H* if there exists $x \in X$, $y \in Y$ such that x and y are connected labels in H.

(2) Let $G = (\Sigma, \Delta, P, C, Z)$ be a NLC grammar.

(2.1) $X, Y \subseteq \Delta$ are *adjacent sets of labels in G* if there exists a graph $H \in L(G)$ such that X and Y are adjacent in H.

(2.2) $X, Y \subseteq \Delta$ are *connected sets of labels in G* if there exists a graph $H \in L(G)$ such that X and Y are connected in H. ▮

II. Two Direct Results

In this section we consider two decision problems whose solutions are rather straightforward (standard). By following those solutions the reader can acquire the basic familiarity with NLC grammars.

Our first theorem provides a positive solution to a quite basic decision problem.

THEOREM 1. *Given an arbitrary NLC grammar $G = (\Sigma, \Delta, P, C, Z)$ and arbitrary labels a, b of Δ, it is decidable whether or not a and b are adjacent in G.*

Proof. We define a sequence $T_0, T_1, T_2, \ldots$ of subsets of $\Sigma \times \Sigma$ as follows:

$$T_0 = \{(x, y) \mid x, y \in \Sigma,\ x \text{ is adjacent to } y \text{ in } Z\};$$

$$T_{i+1} = T_i \cup \{(x, y) \mid x, y \in \Sigma, \text{ and there are } s,\ t \text{ in } T_i \text{ for which there is a graph } M \text{ with } \overset{s}{\bullet}\!\!-\!\!\overset{t}{\bullet} \Rightarrow_G M \text{ and } a \text{ and } b \text{ are adjacent in } M\}.$$

Clearly, if $T_j = T_{j+1}$, we have

$$\begin{aligned} T_{j+2} &= T_{j+1} \cup \{(x, y) \mid x, y \in \Sigma \text{ and there is a pair } (s, t) \text{ in } T_{j+1} \\ &\qquad \text{for which there is a graph } M \text{ with } \overset{s}{\bullet}\!\!-\!\!\overset{t}{\bullet} \Rightarrow_G M \\ &\qquad \text{and } x \text{ and } y \text{ are adjacent in } M\}, \\ &= T_j \cup \{(x, y) \mid x, y \in \Sigma \text{ and there is a pair } (s, t) \text{ in } T_j \\ &\qquad \text{for which there is a graph } M \text{ with } \overset{s}{\bullet}\!\!-\!\!\overset{t}{\bullet} \Rightarrow_G M \\ &\qquad \text{and } x \text{ and } y \text{ are adjacent in } M\}, \\ &= T_{j+1}, \end{aligned}$$

and hence, since $\Sigma \times \Sigma$ is finite, we have for some $j_0 \in \mathbb{N}$: $T_0 \subsetneq T_1 \subsetneq T_2 \subsetneq \cdots \subsetneq T_{j_0} = T_{j_0+1} = T_{j_0+2} = \cdots$. Furthermore, we have: $(x, y) \in T_j$ if and only if there is an H in $S(G)$ for which $Z \Rightarrow_G^j H$ and x is adjacent to y in H. This can be seen as follows:

—If $j=0$, this follows directly from the definition of T_0.

—If $j \neq 0$, then $(x, y) \in T_j$ implies that there is a pair (s, t) in T_{j-1} such that for some graph M, $\overset{s}{\bullet}\!\!-\!\!\overset{t}{\bullet} \Rightarrow_G^1 M$ and x is adjacent to y in M. If the claim holds for $0, 1, \ldots, j-1$, then from $(s, t) \in T_{j-1}$ it follows that there is a graph $\bar{H}$ with $Z \Rightarrow_G^{j-1} \bar{H}$ and $\bar{H}$ has a subgraph of the form $\overset{s}{\bullet}\!\!-\!\!\overset{t}{\bullet}$. Since $\overset{s}{\bullet}\!\!-\!\!\overset{t}{\bullet} \Rightarrow_G^1 M$ it is clear that $\bar{H} \Rightarrow_G^1 H$, where x and y are adjacent in H. On the other hand, if there exists a graph $H \in S(G)$ for which $Z \Rightarrow_G^j H$ and x is adjacent to y in H, then it is easy to see that $(x, y) \in T_j$. This proves the claim. Since the construction of the T's is effective the construction of T_{j_0} is effective and so we have $(a, b) \in T_{j_0}$ if and only if a is adjacent to b in G. ∎

Our second result provides a negative solution to the standard equivalence problem concerning grammars.

THEOREM 2. *Given two arbitrary NLC grammars G_1 and G_2, it is undecidable whether or not $L(G_1) = L(G_2)$.*

Proof. From (16) in Section I it follows that it suffices to show that the equivalence problem is undecidable for RNLC grammars, i.e., it is undecidable whether or not two arbitrary RNLC grammars generate the same language. The RNLC grammar G_1 given below, generates the language

$$\overset{l_1}{\bullet}\!-\!\overset{l_2}{\bullet}\!-\!\overset{l_3}{\bullet}\!-\cdots-\!\overset{l_t}{\bullet}\!-\!\overset{T}{\bullet}\!-\!\overset{\bar{m}_1}{\bullet}\!-\!\overset{\bar{m}_2}{\bullet}\!-\!\overset{\bar{m}_3}{\bullet}\!-\cdots-\!\overset{\bar{m}_r}{\bullet}$$

such that $t, r \geqslant 1$; for $1 \leqslant i \leqslant t$, $l_i \in \{0, 1\}$; for $1 \leqslant j \leqslant r$, $m_j \in \{0, 1\}$; and $l_1 l_2 \cdots l_t \neq$ $\mathbf{mir}(m_1 m_2 \cdots m_r)\}$, where for *a* word x, $\mathbf{mir}(x)$ denotes the mirror image of x.

$$G_1 = (\Sigma_1, \Delta_1, P_1, Z_1),$$

where

$$\Sigma_1 = \{0, 1, \bar{0}, \bar{1}, P, Q, R, S, T\},$$
$$\Delta_1 = \{0, 1, \bar{0}, \bar{1}, T\},$$
$$Z_1 = \overset{S}{\bullet},$$

and if we let $C_{11} = \{0, 1\} \times \{0, 1\}$ and $C_{12} = \{\bar{0}, \bar{1}\} \times \{\bar{0}, \bar{1}\}$, then P_1 has the following productions:

(1) for all $x \in \{0, 1\}$,

$$\left(\left(S, \overset{x}{\bullet}\!-\!\overset{S}{\bullet}\!-\!\overset{\bar{x}}{\bullet}\right); C_{11} \cup C_{12}\right) \in P_1,$$

(2) for all $x \in \{0, 1\}$,

$$\left(\left(S, \overset{x}{\bullet}\!\longrightarrow\!\overset{P}{\bullet}\right); C_{11} \cup \{P\} \times \{\bar{0}, \bar{1}\}\right) \in P_1,$$

$$\left(\left(S, \overset{Q}{\bullet}\!\longrightarrow\!\overset{\bar{x}}{\bullet}\right); C_{12} \cup \{Q\} \times \{0, 1\}\right) \in P_1,$$

$$\left(\left(P, \overset{x}{\bullet}\!\longrightarrow\!\overset{P}{\bullet}\right); C_{11} \cup \{P\} \times \{\bar{0}, \bar{1}\}\right) \in P_1,$$

and

$$\left(\left(Q, \overset{Q}{\bullet}\!\longrightarrow\!\overset{\bar{x}}{\bullet}\right); C_{12} \cup \{Q\} \times \{0, 1\}\right) \in P_1,$$

(3) for all $x, y \in \{0, 1\}$ with $x \neq y$,

$$\left(\left(S, \overset{x}{\bullet}\!\longrightarrow\!\overset{R}{\bullet}\!\longrightarrow\!\overset{\bar{y}}{\bullet}\right) C_{11} \cup C_{12}\right) \in P_1,$$

(4) for all $x \in \{0, 1\}$,

$$\left(\left(R, \overset{x}{\bullet}\!\longrightarrow\!\overset{R}{\bullet}\right); C_{11} \cup \{R\} \times \{\bar{0}, \bar{1}\}\right) \in P_1,$$

$$\left(\left(R, \overset{R}{\bullet}\!\longrightarrow\!\overset{\bar{x}}{\bullet}\right); C_{12} \cup \{R\} \times \{0, 1\}\right) \in P_1,$$

(5) for $x \in \{P, Q, R\}$,

$$\left(\left(x, \overset{T}{\bullet}\right); \{T\} \times \Delta_1\right) \in P_1.$$

If $l_1 l_2 \cdots l_t \neq \mathbf{mir}(m_1 m_2 \cdots m_r)$ then there are two possibilities: either $t \neq r$; this corresponds to a derivation in which productions of type (2) occur, or $t = r$ and for some i, $1 \leqslant i \leqslant t$, $l_i \neq m_{r-i+1}$. This corresponds to a derivation in which productions of type (3) occur. Now let (A, B) be an instance of the post correspondence problem, abbreviated PCP. Let $A = (\alpha_1, \alpha_2, \ldots, \alpha_n)$ and $B = (\beta_1, \beta_2, \ldots, \beta_n)$ be lists of nonempty words over $\{0, 1\}$. We will now construct a RNLC grammar G_2 that generates the language

$\{\overset{l_1}{\bullet}\!\longrightarrow\!\overset{l_2}{\bullet}\!\longrightarrow\!\overset{l_3}{\bullet} \cdots \overset{l_t}{\bullet}\!\longrightarrow\!\overset{p}{\bullet}\!\longrightarrow\!\overset{\bar{m}_1}{\bullet} \cdots \overset{\bar{m}_{r-1}}{\bullet}\!\longrightarrow\!\overset{\bar{m}_r}{\bullet}$ such that there is a sequence $i_1, i_2, \ldots, i_s \in \mathbb{N}$ with $l_1 l_2 \cdots l_t = \alpha_{i_1} \alpha_{i_2} \cdots \alpha_{i_s}$ and $\mathbf{mir}(m_1, m_2, \ldots, m_r) = \beta_{i_1}, \beta_{i_2} \cdots \beta_{i_s}\}$.

Let $G_2 = (\Sigma_2, \Delta_2, P_2, Z_2)$, where

$$\Sigma_2 = \{0, 1, \bar{0}, \bar{1}, 0_{\not c}, 1_{\not c}, \bar{0}_{\not c}, \bar{1}_{\not c}, W, V, T\},$$

$$\Delta_2 = \{0, 1, \bar{0}, \bar{1}, T\},$$

$$Z_2 = \overset{V}{\bullet},$$

and if we let $C_{21} = \{0_{\not c}, 1_{\not c}\} \times \{0, 1, 0_{\not c}, 1_{\not c}\}$ and $C_{22} = \{\bar{0}_{\not c}, \bar{1}_{\not c}\} \times \{\bar{0}, \bar{1}, \bar{0}_{\not c}, \bar{1}_{\not c}\}$, then P_2 has the following productions:

(1) for all $1 \leqslant i \leqslant n$, where $l_1 l_2 \cdots l_t = \alpha_i$ and $\mathbf{mir}(m_1 m_2 \cdots m_r) = \beta_i$,

$$\left(\left(V, \overset{l_{\not c}}{\bullet}\!-\!\overset{l_2}{\bullet} \cdots \overset{l_t}{\bullet}\!-\!\overset{W}{\bullet}\!-\!\overset{\bar{n}_1}{\bullet} \cdots \overset{\bar{n}_{r-1}}{\bullet}\!-\!\overset{\bar{m}_{r\not c}}{\bullet}\right);\ C_{21} \cup C_{22}\right) \in P_2,$$

$$\left(\left(W, \overset{l_{1\not c}}{\bullet}\!-\!\overset{l_2}{\bullet} \cdots \overset{l_t}{\bullet}\!-\!\overset{W}{\bullet}\!-\!\overset{\bar{m}_1}{\bullet} \cdots \overset{\bar{m}_{r-1}}{\bullet}\!-\!\overset{\bar{m}_{r\not c}}{\bullet}\right);\ C_{21} \cup C_{22}\right) \in P_2,$$

and

(2) P_2 includes

$$\left\{\left(\left(0_{\not c}, \overset{0}{\bullet}\right); \{0\} \times \Sigma_2\right), \left(\left(1_{\not c}, \overset{1}{\bullet}\right); \{1\} \times \Sigma_2\right),\right.$$

$$\left.\left(\left(\bar{0}_{\not c}, \overset{\bar{0}}{\bullet}\right); \{\bar{0}\} \times \Sigma_2\right), \left(\left(\bar{1}_{\not c}, \overset{\bar{1}}{\bullet}\right); \{\bar{1}\} \times \Sigma_2\right), \left(\left(W, \overset{T}{\bullet}\right); \{T\} \times \Sigma_2\right)\right\}.$$

Observe that $\Sigma_1 \cap \Sigma_2 = \Delta_1 = \Delta_2$ and neither P_1 nor P_2 contains productions for labels in Σ_1. This makes it easy to construct a RNLC grammar G, with $L(G_3) = L(G_2) \cup L(G_1)$: let $G_3(\Sigma_3, \Delta_3, P_3, Z_3)$, where

$$\Sigma_3 = \Sigma_1 \cup \Sigma_2 \cup \{S'\} \qquad \text{with} \quad S' \notin \Sigma_1 \cup \Sigma_2,$$

$$\Delta_3 = \Delta_1\ (=\Delta_2),$$

$$Z_3 = \overset{S'}{\bullet},$$

$$P_3 = P_1 \cup P_2 \cup \{((S', Z_1); \phi), ((S', Z_2); \phi)\}.$$

It is obvious that $L(G_1) \neq L(G_3)$ if and only if PCP(A, B) has a solution. Since A and B were arbitrary, this proves the theorem. ∎

III. The First Basic Construction

In this section we consider a number of decision problems concerning NLC grammars. The solutions of all of them depend directly or indirectly on the construction that we present next.

We will prove first that it is undecidable whether or not the language of a NLC grammar contains a totally disconnected graph. To this aim we will show that a decision procedure for this problem would lead to a decision procedure for the Post Correspondence problem (PCP). For any instance (A, B) of PCP, we will construct a RNLC grammar $G(A, B)$ such that $L(G(A, B))$ contains a totally disconnected graph if and only if $\mathrm{PCP}(A, B)$ has a solution. By (16) of Section I, this will prove the desired result. Our construction is rather complicated, and for this reason it is presented in several steps.

Step I

We will construct a family $\mathscr{G}_1$ of RNLC grammars that differ from each other by their axioms only. The axioms will be chains, and in each element of $\mathscr{G}_1$ a totally disconnected graph can be derived in exactly one way.

Let σ be the permutation on $\{1, 2, 3\}$defined by $\sigma(1) = 2$, $\sigma(2) = 3$, $\sigma(3) = 1$.

Let $\rho = \sigma^2$ (hence, $\rho(1) = 3$, $\rho(2) = 1$, $\rho(3) = 2$).

An element of $\mathscr{G}_1$ is of the form $(\Sigma_1, \Delta_1, P_1, Z_1)$, where

$$\Sigma_1 = \{a_i \mid i = 1, 2, 3, t\} \cup \{b_i \mid i = 1, 2, 3, t\} \cup \{c_i \mid i = 1, 2, 3, t\} \cup \{d, L\},$$

$$\Delta_1 = \{L, d\},$$

Z_1 is of the form

$$\overset{b_1}{\bullet}\!-\!\overset{a_2}{\bullet}\!-\!\overset{a_3}{\bullet}\!-\!\overset{a_1}{\bullet}\!-\!\overset{a_2}{\bullet}\!-\!\overset{a_3}{\bullet}\!-\!\overset{a_1}{\bullet}\!-\cdots-\!\overset{a_3}{\bullet}\!-\!\overset{a_1}{\bullet}\!-\!\overset{a_2}{\bullet}\!-\!\overset{a_t}{\bullet},$$

P_1 consists of the following productions:

(1a) for $i = 1, 2, 3$: $((a_i, \overset{b_i}{\bullet}\ \overset{L}{\bullet}); \{b_i, L\} \times \Sigma_1 \backslash \{(L, a_{\sigma(i)}), (L, a_t), (L, c_{\rho(i)})\})$,

(1b) $((a_t, \overset{b_t}{\bullet}\ \overset{L}{\bullet}); \{b_t, L\} \times \Sigma_1 \backslash \{(L, c_1), (L, c_2), (L, c_3)\})$,

(2a) for $i = 1, 2, 3$: $((b_i, \overset{c_i}{\bullet}\ \overset{L}{\bullet}); \{c_i, L\} \times \Sigma_1 \backslash \{(L, a_{\sigma(i)}), (L, a_t)\})$,

(2b) $((b_t, \overset{c_t}{\bullet}\ \overset{L}{\bullet}); \{c_t, L\} \times \Sigma_1)$,

(3a) for $i = 1, 2, 3$: $((c_i, \overset{d}{\bullet}\ \overset{L}{\bullet}); \Sigma_1 \times \Sigma_1 \backslash \{(L, b_{\sigma(i)}), (L, b_t), (d, b_{\sigma(i)}), (d, b_t)\})$,

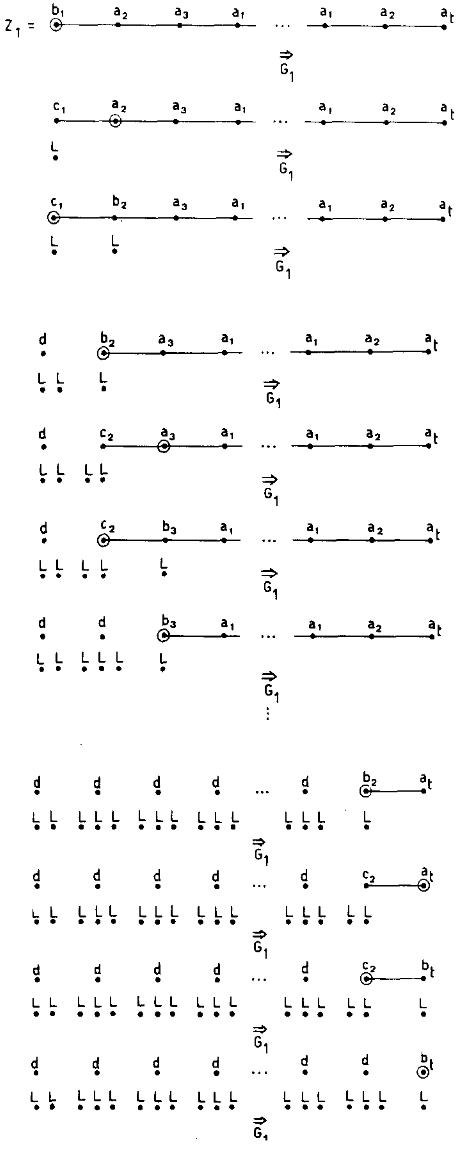
(3b) $((c_t, \overset{d}{\bullet}\ \overset{L}{\bullet}); \Sigma_1 \times \Sigma_1)$,

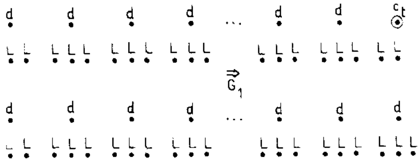
Proposition 1. *Let $G_1 \in \mathscr{G}_1$. If $\bar{H} \Rightarrow^*_{G_1} H$ and H is totally disconnected, then all L-labelled nodes of $\bar{H}$ must be isolated.*

Proof. Assume to the contrary that $\bar{H}$ contains a node n labelled by L and n is not isolated. Let n_1 be a node in $\bar{H}$ adjacent to n. Since $\bar{H} \Rightarrow^*_{G_1} H$, in deriving H from

$\bar{H}$ we must rewrite n_1. (We have no productions for rewriting n.) However, (L, L) is in the connection relation of every production in P_1 and every production in P_1 introduces a node labelled L. Consequently after rewriting n_1, every successively derived graph will contain two adjacent L-labelled nodes, a contradiction.

From the above result it easily follows that, given G_1 in $\mathscr{G}_1$, there exists a unique derivation in G_1 yielding a totally disconnected graph; moreover this derivation is of the following form (we encircle the node that is being rewritten):





Observe that in such a derivation, all intermediate graphs without b-labelled nodes have exactly one c-labelled node.

Step II

Again we will define a family $\mathscr{G}_2$ of RNLC grammars, differing by their axioms only. An element of $\mathscr{G}_2$ is a grammar $(\Sigma_2, \Delta_2, P_2, Z_2)$, where $\Sigma_2 = \bar{\Sigma}_2 \cup \bar{\bar{\Sigma}}_2$, with $\bar{\Sigma}_2 = \{a_i \mid i = 1, 2, 3, t\} \cup \{b_i \mid i = 1, 2, 3, t\} \cup \{c_i \mid i = 1, 2, 3, t\} \cup \{d, L\}$, $\bar{\bar{\Sigma}}_2 = \{A_i \mid i = 1, 2, 3, t\} \cup \{B_i \mid i = 1, 2, 3, t\} \cup \{C_i \mid i = 1, 2, 3, t\} \cup \{D, L\}$ (obviously we require that $\bar{\Sigma}_2 \cap \bar{\bar{\Sigma}}_2 = \{L\}$),

$$\Delta_2 = \{d, D, L\},$$

Z_2 is of the form shown in Fig. 1 (where in general there is no relation between the lengths of the "upper" and the "lower" chain and all nodes of the upper chain are adjacent to all nodes of the lower chain), P_2 consists of the following productions (let σ, ρ be defined as in I above):

(1a) for $i = 1, 2, 3$: $((a_i, {}^{b_i\,L}_{\bullet\,\bullet}); \{b_i, L\} \times \bar{\Sigma}_2 \setminus \{(L, a_{\sigma(i)}),\ (L, a_t), (L, c_{\rho(i)})\}$
$\cup \{(L, C_1), (L, C_2), (L, C_3), (L, C_t)\} \cup \{b_i\} \times \bar{\bar{\Sigma}}_2)$,

(1b) $((a_t, {}^{b_t\,L}_{\bullet\,\bullet}); \{b_t, L\} \times \bar{\Sigma}_2 \setminus \{(L, c_1), (L, c_2), (L, c_3)\}$
$\cup \{(L, C_1), (L, C_2), (L, C_3), (L, C_t)\} \cup \{(L, A_t)\} \cup \{b_t\} \times \bar{\bar{\Sigma}}_2)$,

(2a) for $i = 1, 2, 3$: $((b_i, {}^{c_i\,L}_{\bullet\,\bullet}); \{c_i, L\} \times \bar{\Sigma}_2 \setminus \{(L, a_{\sigma(i)}), (L, a_t)\}$
$\cup \{(L, B_1), (L, B_2), (L, B_3), (L, B_t)\} \cup \{c_i\} \times \bar{\bar{\Sigma}}_2)$,

(2b) $((b_t, {}^{c_t\,L}_{\bullet\,\bullet}); \{c_t, L\} \times \bar{\Sigma}_2 \cup \{(L, B_1), (L, B_2), (L, B_3), (L, B_t)\}$
$\cup \{c_t\} \times \bar{\bar{\Sigma}}_2 \cup \{(L, C_t)\})$,

(3a) for $i = 1, 2, 3$: $((c_i, {}^{d\,L}_{\bullet\,\bullet});$
$\{d, L\} \times \Sigma_2 \setminus \{(L, b_{\sigma(i)}), (L, b_t), (d, b_{\sigma(i)}), (d, b_t)\}$
$\cup \{(L, C_1), (L, C_2), (L, C_3), (L, C_t)\})$,

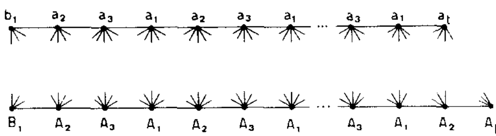


FIG. 1. General form of Z_2.

(3b) $((c_t, \overset{d}{\bullet}\quad \overset{L}{\bullet}); \{d, L\} \times \bar{\Sigma}_2 \cup \{(L, C_1), (L, C_2), (L, C_3), (L, C_t)\})$,

(4a) for $i = 1, 2, 3$: $((A_i, \overset{B_i}{\bullet}\overset{L}{\bullet}); \{B_i, L\} \times \bar{\bar{\Sigma}}_2 \backslash \{(L, A_{\sigma(i)}), (L, A_t), (L, C_{\rho(i)})\}$
$\cup \{(L, b_1), (L, b_2), (L, b_3), (L, b_t)\} \cup \{B_i) \times \bar{\Sigma}_2)$,

(4b) $((A_t, \overset{B_t}{\bullet}\overset{L}{\bullet}); \{B_t, L\} \times \bar{\bar{\Sigma}}_2 \backslash \{(L, C_1), (L, C_2), (L, C_3)\}$
$\cup \{(L, b_1), (L, b_2), (L, b_3), (L, b_t)\} \cup \{(L, a_1), (L, a_2), (L, a_3)\}$,
$\cup \{B_t\} \times \bar{\Sigma}_2$,

(5a) for $i = 1, 2, 3$: $((B_i, \overset{C_i}{\bullet}\overset{L}{\bullet}); \{C_i, L\} \times \bar{\bar{\Sigma}}_2 \backslash \{(L, A_{\sigma(i)}), (L, A_t)\}$
$\cup \{(L, c_1), (L, c_2), (L, c_3), (L, c_t)\} \cup \{C_i\} \times \bar{\Sigma}_2)$,

(5b) $((B_t, \overset{C_t}{\bullet}\overset{L}{\bullet}); \{C_t, L\} \times \bar{\bar{\Sigma}}_2 \cup \{(L, c_1), (L, c_2), (L, c_3), (L, c_t)\} \cup \{C_t\} \times \bar{\Sigma}_2)$,

(6a) for $i = 1, 2, 3$: $((C_i, \overset{D}{\bullet}\overset{L}{\bullet});$
$\{D, L\} \times \bar{\bar{\Sigma}}_2 \backslash \{(L, B_{\sigma(i)}), (L, B_t), (D, B_{\sigma(i)}), (D, B_t)\})$,

(6b) $((C_t, \overset{D}{\bullet}\overset{L}{\bullet}); \{D, L\} \times \bar{\bar{\Sigma}}_2)$.

PROPOSITION 2. *If $G_2 = (\Sigma_2, \Delta_2, P_2, Z_2) \in \mathscr{G}_2$, then a totally disconnected graph can be derived in G_2 if and only if the upper and the lower chains of the axiom Z_2 have equal lengths. If a totally disconnected graph can be derived, then it is unique and it has a unique derivation in G_2.*

Proof. (a) Proposition 1 of I holds for G_2 also: if $\bar{H} \Rightarrow^*_{G_2} H$, H is totally disconnected, then all L-labelled nodes of $\bar{H}$ are isolated.

(b) Let $\bar{h}$, $\bar{\bar{h}}$ be weak codings defined by

$$\bar{h}(x) = x \quad \text{if} \quad x \in \bar{\Sigma}_2, \qquad \bar{h}(x) = \Lambda \quad \text{if} \quad x \in \bar{\bar{\Sigma}}_2 \backslash \{L\},$$
$$\bar{\bar{h}}(x) = x \quad \text{if} \quad x \in \bar{\bar{\Sigma}}_2, \qquad \bar{\bar{h}}(x) = \Lambda \quad \text{if} \quad x \in \bar{\Sigma}_2 \backslash \{L\}.$$

Let $\bar{P}_2$ be the set of productions, resulting from the sets of productions (1a), (1b), (2a), (2b), (3a), (3b) from the definition of P_2 by replacing each connection relation by its intersection with $\bar{\Sigma}_2 \times \bar{\Sigma}_2$. Let $\bar{\bar{P}}_2$ be the set of productions, resulting from the sets of productions (4a), (4b), (5a), (5b), (6a), (6b) by replacing each connection relation by its intersection with $\bar{\bar{\Sigma}}_2 \times \bar{\bar{\Sigma}}_2$. Then obviously, $(\bar{\Sigma}_2, \{d, L\}, \bar{P}, \bar{h}(Z_2)) \in \mathscr{G}_1$ and if K is the RNLC system, resulting from $(\bar{\bar{\Sigma}}_2, \{D, L\}, \bar{\bar{P}}, \bar{\bar{h}}(Z_2))$ by replacing all capital letters (except L) by small letters, then $K \in \mathscr{G}_1$.

(c) If $H \in S(G_2)$,

$$n_1 \in V_H, \varphi_H(n_1) \in \bar{\Sigma}_2 \backslash \{d, L\},$$
$$n_2 \in V_H, \varphi_H(n_2) \in \bar{\bar{\Sigma}}_2 \backslash \{D, L\},$$

then $\{n_1, n_2\} \in E_H$. (Informally: all a-, b- or c-labelled nodes are adjacent to all A-, B- or C-labelled nodes.)

Proof of (c) Suppose $\{n_1, n_2\} \notin E_H$. Let $\bar{H}$ be the last graph of a derivation sequence for H in which the ancestors n_1', n_2' of n_1 and n_2 are connected. Then, in the following step, either a production of one of the forms (1a), (1b), (2a), (2b), is applied to n_1' or a production of one of the forms (4a), (4b), (5a), (5b) is applied to n_2'. It is easily seen that all those cases lead to a contradiction; e.g., the connection graph of (1a) contains $\{b_i\} \times \bar{\bar{\Sigma}}_2$.

(d) From (a) and (c) it follows that a derivation leading to a totally disconnected graph must satisfy the following restrictions:

—a production of the form (1a) or (1b) may not be applied to a graph that contains a C_α-labelled node ($\alpha \in \{1, 2, 3, t\}$),

—a production of the form (2a) or (2b) may not be applied to a graph that contains a B_α-labelled node ($\alpha \in \{1, 2, 3, t\}$),

—a production of the form (3a) or (3b) may not be applied to a graph that contains a C_α-labelled node ($\alpha \in \{1, 2, 3, t\}$),

—a production of the form (4a) or (4b) may not be applied to a graph that contains a b_α-labelled node ($\alpha \in \{1, 2, 3, t\}$),

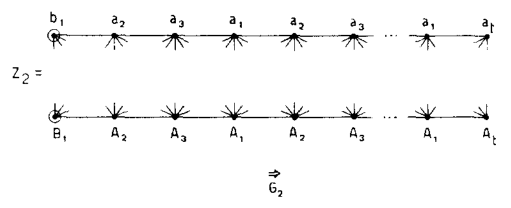
—a production of the form (5a) or (5b) may not be applied to a graph that contains a c_α-labelled node ($\alpha \in \{1, 2, 3, t\}$),

–a production of the form (1b) may not be applied to a graph that contains an A_t-labelled node,

—a production of the form (4b) may not be applied to a graph that contains an a_α-labelled ($\alpha \in \{1, 2, 3\}$),

—a production of the form (2b) may not be applied to a graph that contains a C_t-labelled node.

(e) Now it follows from (b) and (d) that the only derivation leading to a totally disconnected graph is of the form depicted in Figs. 2a and 2b. (In each step, it follows from (b) that we must rewrite one of the encircled nodes.)



(By (d) we know that only a production of type (5a) may be applied.)

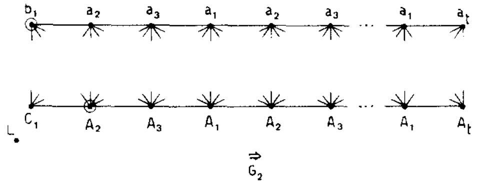
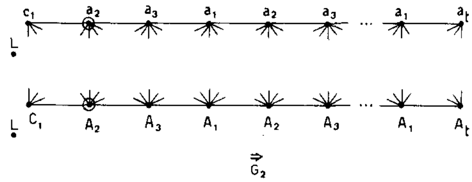
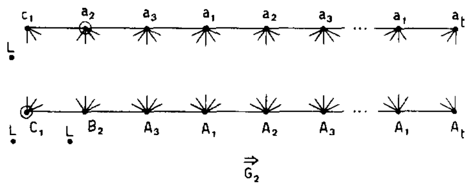


FIG. 2a. Beginning of a derivation in G_2 leading to a totally disconnected graph.

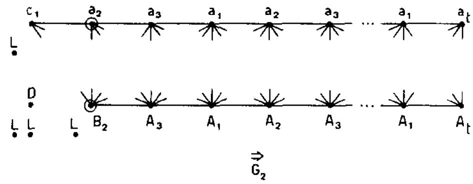
(From (d) it follows that the "upper" node must be rewritten, because productions of the form (4a) may not be applied.)



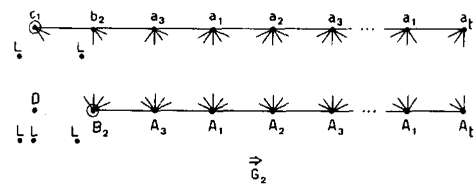
(Since, by (d), a production of type (1a) cannot be applied, the lower node must be rewritten.)



(From (d) we know that a production of type (1a) cannot be applied.)



(From (d) it follows that we may not use a production of the form (5a).)



(From (d) follows that we may not use a production of the form (5a).)

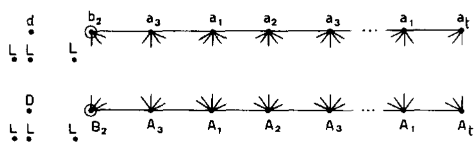


FIG. 2a—*Continued.*

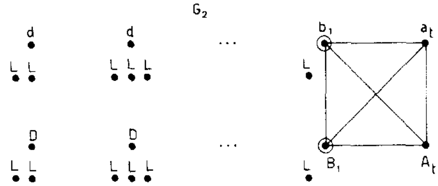
(Thus, a "macrostep" was completed and the rest of the derivation iterates this macrostep, "breaking" connections from the left to right.)

$$\underset{G_2}{\Rightarrow}$$

$$\vdots \; .$$

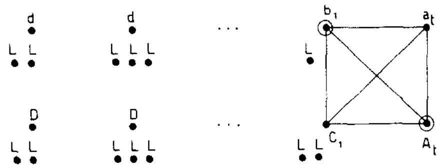
FIG. 2a—*Continued.*

$$\cdots \underset{G_2}{\Rightarrow}$$



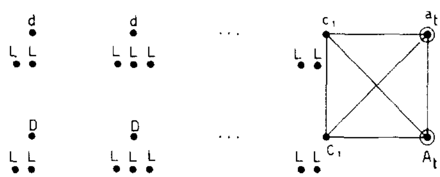
$$\underset{G_2}{\Rightarrow}$$

(From (d) it follows that productions of the form (2a) may not be used.)



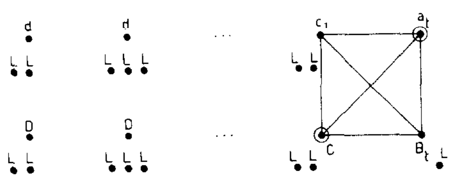
$$\underset{G_2}{\Rightarrow}$$

(From (d) it follows that we cannot use a production of type (4b).)



$$\underset{G_2}{\Rightarrow}$$

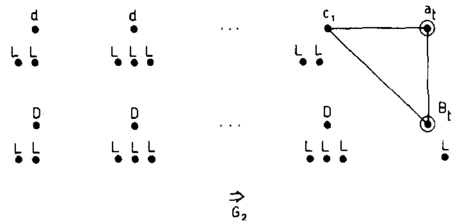
(From (d) we know that we may not apply a production of type (1b).)



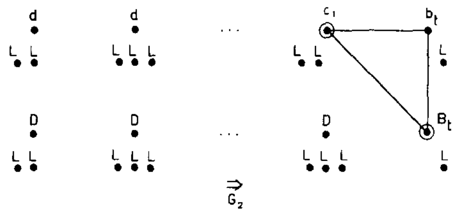
$$\underset{G_2}{\Rightarrow}$$

FIG. 2b. Last part of the derivation from FIG. 2a.

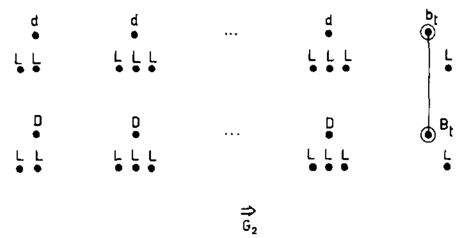
(From (d) we know that we may not apply a production of type (1b).)



(From (d) we know that we may not apply a production of type (5b).)



(From (d) it follows that we may not ue a production of type (5b) so we must rewrite the upper one.)



(It follows from (d) that we must rewrite the lower, encircled node.)

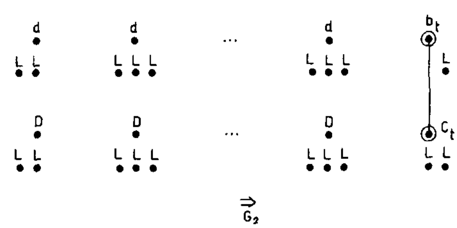
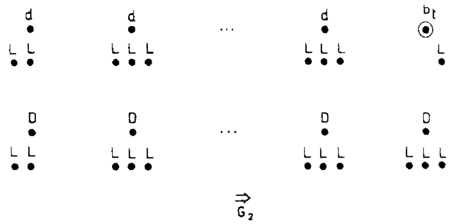


FIG. 2b—*Continued.*

(Again, it follows from (d) that we cannot rewrite the upper, encircled node.)



(From now on, only one node has a label for which there is a production).

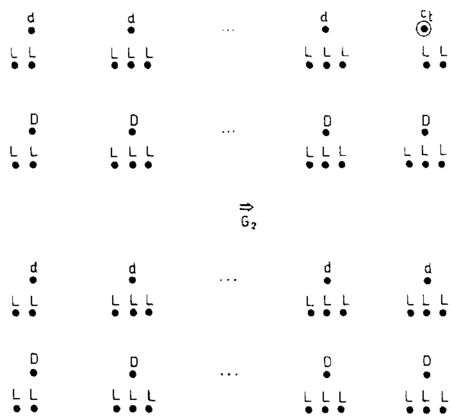


FIG. 2b—*Continued.*

The last graph is an element of $L(G_2)$.

(f) The connections between the upper and the lower chains are "broken" by applying productions of type (6a) or (6b). Formally, if $Z_2 \Rightarrow^*_{G_2} \bar{H} \Rightarrow^1_{G_2} \bar{\bar{H}} \Rightarrow^*_{G_2} H$, H is totally disconnected, $n_1, n_2 \in V_{\bar{\bar{H}}}$, $\varphi_{\bar{\bar{H}}}(n_1) \in \bar{\Sigma}_2$, $\varphi_{\bar{\bar{H}}}(n_2) \in \bar{\Sigma}_2$, $\{n_1, n_2\} \notin E_{\bar{\bar{H}}}$, the ancestors of n_1 and n_2 in $\bar{H}$ are adjacent, then $\bar{\bar{H}}$ is derived from $\bar{H}$ by applying a production of type (6a) or (6b) to one of those ancestors.

(g) If G_2 has an axiom Z_2 in which the upper chain and the lower chain are of different length, no totally disconnected graph can be derived in G_2.

Proof. From (a) through (d) it follows that a derivation of a totally disconnected graph must be as illustrated in (e).

(1) If the upper chain of Z_2 is longer than the lower one (Fig. 3), then we arrive at the situation of Fig. 4, where no further productions can be applied without deriving a graph that is not totally disconnected.

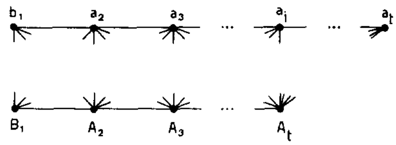


FIGURE 3

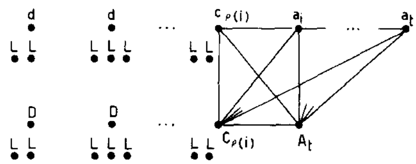


FIGURE 4

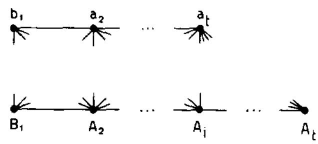


FIGURE 5

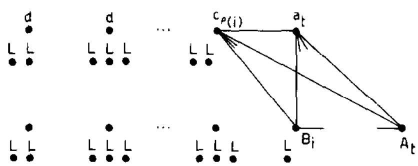


FIGURE 6

(2) If the lower chain of Z_2 is longer (Fig. 5), then we reach the situation of Fig. 6, where again no further production can be applied without deriving a graph that is not totally disconnected. ∎

Step III

We now define a family of RNLC-grammars $\mathscr{G}_3$, that is similar to $\mathscr{G}_2$ except that positive integers are added as second components of the labels. Let n be an arbitrary, but fixed, positive integer. Then the elements of $\mathscr{G}_3$ are of the form

$$(\Sigma_3, \Delta_3, P_3, Z_3),$$

where

$$\Sigma_3 = \Sigma_2 \times \{1, \ldots, n\},$$

$$\Delta_3 = \Delta_2 \times \{1, \ldots, n\},$$

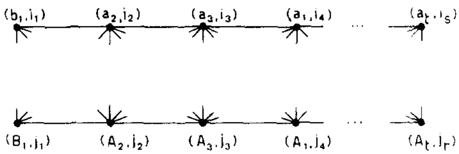


FIG. 7. General form of Z_3

Z_3 is of the form shown in Fig. 7 (s does not have to be equal to r) and P_3 consists of the following productions:

(1) For each production π of one of the types (1a), (1b), (2a), (2b), (3a), (3b), (4a), (4b), (5a), (5b):

$$\pi = \left(\left(X_\alpha, \begin{smallmatrix} Y_\alpha & L \\ \bullet & \bullet \end{smallmatrix} \right); C(\pi) \right).$$

P_3 includes, for every $j \in \{1,..., n\}$, the production

$$\left(\left((X_\alpha, j), \begin{smallmatrix} (Y_\alpha, j) & (L, j) \\ \bullet & \bullet \end{smallmatrix} \right); \{((l, k), (l', k')) \mid (l, l') \in C(\pi)\} \right).$$

(2) For each production π of (6a), (6b),

$$\pi = \left(\left(C_\alpha, \begin{smallmatrix} D & L \\ \bullet & \bullet \end{smallmatrix} \right); C(\pi) \right).$$

Let P_3 include, for every $j \in \{1,..., n\}$, the production

$$\left(\left((C, j), \begin{smallmatrix} (D, j) & (L, j) \\ \bullet & \bullet \end{smallmatrix} \right); \{((l, k), (l', k')) \mid (l, l') \in C(\pi)\} \right.$$

$$\left. \cup \{((L, k), (C_m, k')) \mid k \neq k', m \in \{1, 2, 3, t\}\} \right).$$

PROPOSITION 3. *If $G_3 \in \mathscr{G}_3$, then a totally disconnected graph can be derived in G_3 in at most one way, and only under the condition that*

(1) *the upper and lower chains of the axiom are of the same length;*

(2) *the sequences of second components, read from left to right, for both the upper and lower chains are identical.*

Proof. (1) Let h be the coding from Σ_3 to Σ_2 defined by $h(x, i) = x$. Then the definition of P_3 above implies that if $H \Rightarrow_{G_3} H'$ by applying a production corresponding to the forms (1a), (1b), (2a), (2b), (3a), (3b), (4a), (4b), (5a) or (5b), then $h(H) \Rightarrow_{G_2} h(H')$. Consequently, statements analogous to (a) through (g) of the proof of Proposition 2 hold for G_3. Thus, (1) holds.

(2) The definition of P_3 given above implies that in a derivation of a totally disconnected graph, a production corresponding to (6a) or (6b) from P_2 may be applied to the ith node (counted from left to right) of the lower chain of an intermediate graph only if the second component of its label equals the second component of the label of the ith node in the upper chain of this graph. Since the proof of (1) above implies the uniqueness of the derivation of a totally disconnected graph, this implies (2). ∎

Step IV

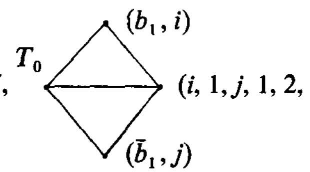
Let $(A = (\alpha_1, \alpha_2, ..., \alpha_n),\ B = (\beta_1, \beta_2, ..., \beta_n))$ be an instance of PCP. We will now give a grammar $G_4(A, B)$ that will generate, among others, graphs of the form that the axioms of the grammars of $\mathscr{G}_3$ had. Let $N = \{1, 2, ..., n\}$, and let $m = \max_{i\in N} |\alpha_i|$, $f = \max_{i\in N} |\beta_i|$. Let $M = \{1, 2, ..., m\}$ and $F = \{1, 2, ..., f\}$. Let σ, ρ be defined as in step I, and for $1 \leqslant j \leqslant |\alpha_i|$, $1 \leqslant k \leqslant |\beta_i|$ let $\alpha(i, j)$ and $\beta(i, k)$ denote the jth and the kth letter of α_i and β_i, respectively.

Then let $G_4(A, B) = (\Sigma_4, \Delta_4, P_4, Z_4)$ be a RNLC grammar such that

$$\Sigma_4 = \Gamma_4 \cup \bar{\Gamma}_4 \cup \bar{\bar{\Gamma}}_4 \cup \{L, Z, T_0, T_1, T_2, T_3, T_t\},$$
$$\text{where} \quad \Gamma_4 = \Sigma_3 \text{ (with } n = \#A = \#B),$$
$$\bar{\Gamma}_4 = \{(\bar{x}, j) \mid (x, j) \in \Sigma_3\}, \qquad \bar{\bar{\Gamma}}_4 = N \times M \times N \times F \times \{1, 2, 3\} \times \{1, 2, 3\},$$
$$\Delta_4 = \{L\} \cup \Gamma_4 \cup \bar{\Gamma}_4,$$
$$Z_4 = \overset{Z}{\bullet}\ ;$$

P_4 consists of the following productions:

(1) For all $i, j \in N$ such that $\alpha(i, 1) = \beta(j, 1)$,

$\Bigg(Z,$ 

$;\ \Sigma_4 \times \Sigma_4\Bigg).$

(2) For all $i, j \in N$, $l \in M$, $k \in F$, r, $s \in \{1, 2, 3\}$ such that $|\alpha_i| > l$ and $|\beta_j| > k$, and $\alpha(i, l+1) = \beta(j, k+1)$,

$$\Big((i, l, j, k, r, s),\ \overset{(i,\, l+1,\, j,\, k+1,\, r,\, s)}{\bullet}\ ;\ \Sigma_4 \times \Sigma_4\Big) \in P_4.$$

(3) For all $i, i', j \in N$, $k \in F, r, s \in \{1, 2, 3\}$ with $|\beta_j| > k$ and $\alpha(i', 1) = \beta(j, k+1)$,

$$\left(\left((i, |\alpha_i|, j, k, r, s), \begin{array}{l} (a_r, i') \\ \big| \\ (i', 1, j, k+1, \sigma(r), s) \end{array}\right);\right.$$

$$\left.\bar{\bar{\Gamma}}_4 \times (\bar{\Gamma}_4 \cup \{T_0, L\}) \cup \{(a_r, i')\} \times (\Gamma_4 \cup \{T_0, L\})\right) \in P_4.$$

(4) For all $i, j, j' \in N$, $l \in M$, $r, s \in \{1, 2, 3\}$ with $|\alpha_i| > l$ and $\alpha(i, l+1) = \beta(j', 1)$,

$$\left(\left((i, l, j, |\beta_j|, r, s), \begin{array}{l} (i, l+1, j', 1, r, \sigma(s)) \\ \big| \\ (\bar{a}_s, j') \end{array}\right);\right.$$

$$\left.\bar{\bar{\Gamma}}_4 \times (\Gamma_4 \cup \{T_0, L\}) \cup \{(\bar{a}_s, j')\} \times (\bar{\Gamma}_4 \cup \{T_0, L\})\right) \in P_4.$$

(5) For all $i, i', j, j' \in N$, $r, s \in \{1, 2, 3\}$, with $\alpha(i', 1) = \beta(j', 1)$,

$$\left(\left((i, |\alpha_i|, j, |\beta_j|, r, s), \begin{tikzcd} (a_r, i') \arrow[dr] & \\ & (i', 1, j', 1, \sigma(r), \sigma(s)) \\ (\bar{a}_s, j') \arrow[ur] & \end{tikzcd}\right)\right.$$

$$\left.\bar{\bar{\Gamma}}_4 \times \{T_0, L\} \cup \{(a_r, i')\} \times (\Gamma_4 \cup \{T_0, L\}) \cup \{(\bar{a}_s, j')\} \times (\bar{\Gamma}_4 \cup \{T_0, L\})\right) \in P_4.$$

(6) For all $i, j, k \in N$, $r, s \in \{1, 2, 3\}$,

$$\left(\left((i, |\alpha_i|, j, |\beta_j|, r, s), \begin{array}{l} (a_t, k) \\ \big| \\ (\bar{a}_t, k) \end{array}\right);\right.$$

$$\left.\{(a_t, k)\} \times (\Gamma_4 \cup \{T_0, L\}) \cup \{(\bar{a}_t, k)\} \times (\bar{\Gamma}_4 \cup \{T_0, L\})\right) \in P_4.$$

(7) For all $j \in N$,

$$\left(\left(\left(T_0, \begin{tikzcd} (B_1, j) \arrow[dr] & \\ L & T_1 \\ (\bar{B}_1, j) \arrow[ur] & \end{tikzcd}\right);\right.\right.$$

$$\left.\{(B_1, j)\} \times \Gamma_4 \cup \{(\bar{B}_1, j)\} \times \bar{\Gamma}_4 \cup \{T_1\} \times (\Gamma_4 \cup \bar{\Gamma}_4) \cup \{L\} \times \bar{\bar{\Gamma}}_4\right) \in P_4.$$

(8) For all $j \in N$,

$$\left(\left(T_0, \begin{array}{c} \cdot \searrow^{(B_1,j)} \\ \overset{L}{\cdot} \qquad \rightarrow T_t \\ \cdot \nearrow_{(\bar{B}_1,j)} \end{array}\right);\right.$$

$$\left.\{(B_1,j)\} \times \Gamma_4 \cup \{(\bar{B}_1,j)\} \times \bar{\Gamma}_4 \cup \{T_t\} \times (\Gamma_4 \cup \bar{\Gamma}_4) \cup \{L\} \times \bar{\bar{\Gamma}}_4\right) \in P_4.$$

(9) For $i = 1, 2, 3$, for all $j \in N$,

$$\left(\left(T_i, \begin{array}{c} \cdot \searrow^{(A_{\sigma(i)},j)} \\ \qquad \rightarrow T_{\sigma(i)} \\ \cdot \nearrow_{(\bar{A}_{\sigma(i)},j)} \end{array}\right);\right.$$

$$\left.\{(A_{\sigma(i)},j)\} \times \Gamma_4 \cup \{(\bar{A}_{\sigma(i)},j)\} \times \bar{\Gamma}_4 \cup \{T_{\sigma(i)}\} \times (\Gamma_4 \cup \bar{\Gamma}_4)\right) \in P_4.$$

(10) For $i = 1, 2, 3$, for all $j \in N$,

$$\left(\left(T_i, \begin{array}{c} \cdot \searrow^{(A_{\sigma(i)},j)} \\ \qquad \rightarrow T_t \\ \cdot \nearrow_{(\bar{A}_{\sigma(i)},j)} \end{array}\right);\right.$$

$$\left.\{(A_{\sigma(i)},j)\} \times \Gamma_4 \cup \{(\bar{A}_{\sigma(i)},j)\} \times \bar{\Gamma} \cup \{T_t\} \times (\Gamma_4 \cup \bar{\Gamma}_4)\right) \in P_4.$$

(11) For all $j \in N$,

$$\left(\left(T_i, \begin{array}{l} \cdot\, (A_t,j) \\ \cdot\, (\bar{A}_t,j) \end{array}\right); \{(A_t,j)\} \times \Gamma_4 \cup \{(\bar{A}_t,j)\} \times \Gamma_4\right) \in P_4.$$

Observe that there are no productions for elements of Δ_4.

Proposition 4. *$L(G_4(A,B))$ contains all graphs of the form shown in Fig. 8, where $\alpha_{i_1}\alpha_{i_2}\cdots\alpha_{i_{s-1}} = \beta_{k_1}\beta_{k_2}\cdots\beta_{k_{q-1}}$, $i_s = k_q$, $r, s, q \geqslant 2$ and all the i's and k's are in N and moreover, every graph in $L(G_4(A,B))$ that is not of this form contains a non-isolated L-labelled node.*

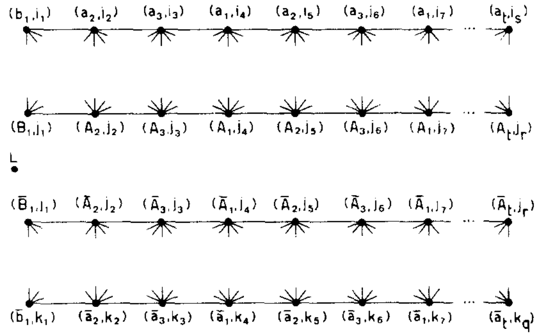


FIG. 8. General form of the graphs in $L(G_4(A, B))$ without non-isolated L-labelled node.

Proof. (a) In any derivation of a graph of $L(G_4(A, B))$, there is precisely one step in which a production of the form (7) or (8) is applied.

Proof of (a). It is easy to see that any derivation in $G_4(A, B)$ has to start with a production of form (1). Since $T_0 \notin \Delta_4$, the T_0-labelled node has to be rewritten at some step. But productions of types (7) and (8) are the only productions, rewriting T_0. Hence, a step as in the statement of (a) exists.

To prove that precisely one such step exists we notice that productions of the form (1) are the only productions, introducing T_0-labelled nodes. Since no rewriting ever introduces a Z-labelled node, a T_0-labelled node is introduced only once. ∎

(b) Suppose $\mathscr{D}$ is a derivation of length v, resulting in a graph of the form of Fig. 8, i.e., a graph without a non-isolated L-labelled node. Let w be the number of the step where a production of type (7) or (8) is applied. Then in steps 1 up to $(w-1)$ only productions of types (1) through (6) are applied, and in steps w up to v only productions of types (7) through (11) are applied.

Proof of (b). Since (7) and (8) are the only types of productions that introduce a T_i-labelled node (where $i = 1, 2, 3, t$), it is clear that productions of types (9), (10), (11) cannot be applied before the $w + 1$th step. It now follows from (a) above that it is enough to prove that productions of types (1) through (6) do not occur in $\mathscr{D}$ after the wth step. We prove this by contradiction. Suppose a production of type (1) through (6) is applied after the wth step. It is clear from the definition of G_4 that no element of $S(G_4(A, B))$ has more than one node with label in $\overline{\overline{\Gamma}}_4$, and that the only graph without a node with label in $\overline{\overline{\Gamma}}_4$ that derives a graph with such a node is the axiom. We conclude that only productions of types (1) through (5) are used before step w. This implies that the graph H_{w-1}, resulting from the $(w-1)$th step in $\mathscr{D}$, has a node m with label in $\overline{\overline{\Gamma}}_4$. It follows from the definition of types (1) through (5) that in H_{w-1}, m is adjacent to the T_0-labelled node n. The rewriting of n in the wth step, using a production of type (7) or (8), results in a graph containing an L-labelled node that is adjacent to m. It follows from the definition of G_4 that no totally disconnected graph can be derived from this graph: an L-labelled node cannot be rewritten and the

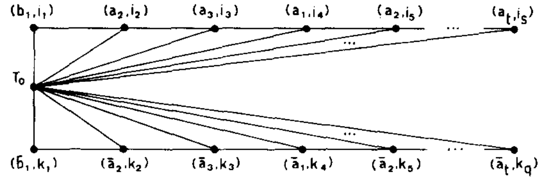


FIG. 9. General form of H_{w-1}.

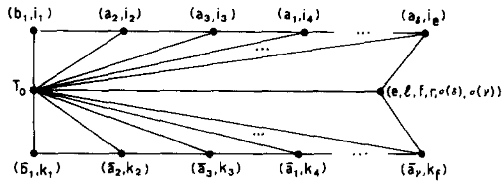


FIG. 10. General form of the H_i's.

rewriting of m leads to at least one descendant $\bar{m}$ that has a label in $\Gamma_4 \cup \bar{\Gamma}_4$ and that is adjacent to the L-labelled node. Since $\bar{m}$ cannot be rewritten, this is a contradiction. ∎

(c) We now prove that H_{w-1} defined in (b) is of the form shown in Fig. 9, with $\alpha_{i_1}\alpha_{i_2}\cdots\alpha_{i_{s-1}} = \beta_{k_1}\beta_{k_2}\cdots\beta_{k_{q-1}}$ and $i_s = k_q$.

Proof of (c). Clearly, in the first step a production of type (1) is applied, and in step $(w-1)$, a production of type (6) is used. We let H_i, $0 \leqslant i \leqslant v$ denote the graph, resulting from the ith step of $\mathscr{D}$. We now prove by induction on the number of steps, i, that for $1 \leqslant i \leqslant w-2$, H_i is of the form shown in Fig. 10, where $\gamma, \delta \in \{1, 2, 3\}$ and $\alpha_{i_1}\alpha_{i_2}\cdots\alpha_{i_{e-1}}\alpha(i_e, 1)\,\alpha(i_e, 2)\cdots\alpha(i_e, l) = \beta_{k_1}\beta_{k_2}\cdots\beta_{k_{f-1}}\beta(k_f, 1)\,\beta(k_f, 2)\cdots\beta(k_f, r)$.

—For $i = 1$, this property follows immediately from the definition of productions of type (1).

—Since a production of type (6) can only be applied in step $w-1$, it is sufficient to show that the application of a production of type (2) through (5) to a graph of this form gives again a graph of this form. This follows immediately from the definition of those productions.

It follows from the definition of the productions of type (6) that H_{w-1} is of the desired form. ∎

(d) From the definition of G_4 it follows, by induction on the length of the word $\alpha_{i_1}\alpha_{i_2}\cdots\alpha_{i_{e-1}}\alpha(i_e, 1)\ \alpha(i_e, 2)\cdots\alpha(i_e, l)$, that all graphs of the form (*) can be derived, using only productions of the forms (1) through (6).

Combining (b), (c) and (d), it follows from the definition of the productions of types (7) through (11) that $L(G_4(A, B))$ is indeed of the form described in Proposition 4. ∎

Using the construction above we can prove now an undecidability result which while interesting on its own will also be useful in providing other undecidable properties of NLC grammars.

THEOREM 3. *It is undecidable whether or not, given a NLC grammar G, $L(G)$ contains a totally disconnected graph.*

Proof. Let $(A = (\alpha_1, \alpha_2, \ldots, \alpha_n), B = (\beta_1, \beta_2, \ldots, \beta_n))$ be an instance of PCP. Let G_3 be constructed as in part III of the above construction (set $n = \#A = \#B$), and let $G_4(A, B)$ be constructed as in part IV of the above construction. Let $G(A, B)$ be the RNLC grammar $(\Sigma, \Delta, P_3, \hat{Z})$ defined as follows.

(i) $\Sigma = \Sigma_4 \cup \{L, Q\}$, where $\bar{Q}, Q \notin \Sigma_4$,

$$\Delta = \Delta_3 \cup \bar{\Sigma}_3 \cup \{L, Q\} \text{ with } \bar{\Delta}_3 = \{(\bar{x}, j) \mid (x, j) \in \Delta_3\},$$

$$\hat{Z} = \underset{n_1}{\overset{\bar{Q}}{\bullet}}\!\!-\!\!-\!\!-\!\!\underset{n_2}{\overset{Z}{\bullet}}$$

(ii) Let $\bar{P}_3$ result from P_3 by replacing the labels from Γ_4 by their counterparts from $\bar{\Gamma}_4$. Add the set $\Sigma \times \{\bar{Q}, Q, L\}$ to the connection relation of each production from P_3 and $\bar{P}_3$, and let $\tilde{P}_3$ and $\tilde{\bar{P}}_3$ respectively be the resulting sets of productions. Add the set $\Sigma \times \{\bar{Q}, Q\}$ to the connection relation of each production from P_4; let $\tilde{P}_4$ be the resulting set of productions. Now add an isolated Q-labelled node to the right-hand side of each production from $\tilde{P}_3$ and $\tilde{\bar{P}}_3$; let Π_3 and $\bar{\Pi}_3$ be the so obtained sets of productions. Finally let

$$P = P_4 \cup \Pi_3 \cup \bar{\Pi}_3 \cup \left\{ \left(\bar{Q}, \overset{Q}{\bullet} \right); \Sigma \times (\Sigma \backslash \Delta_4) \right\}.$$

Let $\mathscr{D}$ be a derivation of a totally disconnected graph. Clearly, in $\mathscr{D}$ the production $(\bar{Q}, \overset{Q}{\bullet})$ must be applied once. Since Q cannot be rewritten, and all connection relations contain $\Sigma \times \{Q\}$, this production must be applied to a graph H labelled by elements of $\Delta_4 \cup \{\bar{Q}\}$ only. In $\mathscr{D}$ no production of $\Pi_3 \cup \bar{\Pi}_3$ can be applied before $\bar{Q}$ is rewitten, because the application of these productions results in a graph $\bar{H}$ that contains an edge $\overset{Q}{\bullet}\!\!-\!\!-\!\!\overset{\bar{Q}}{\bullet}$. Obviously no totally disconnected graph can be derived from $\bar{H}$. Consequently, if m is the $\bar{Q}$-labelled node of H, then $M \backslash m \in L(G_4(A, B))$. Now suppose that, in $H \backslash m$, the L-labelled node is adjacent to another node, v say. Then from the definition of P it follows that in every graph, derived from H, all descendants of v are adjacent to the L-labelled node. This is a contradiction. Thus we conclude that $H \backslash m$ is of the first form from the statement of Proposition 4. Moreover, it is easily seen that no production of $\tilde{P}_4$ can be applied to a graph that is derived from H.

The above conclusions, combined with Propositions 3 and 4, yield that a totally disconnected graph can be derived in $G(A, B)$ if and only if PCP(A, B) has a solution. ∎

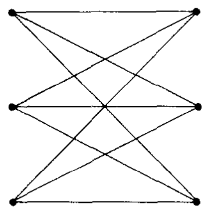


FIG. 11. The graph K_6.

One of the central notions of graph theory is the notion of a planar graph. Hence it is quite natural to investigate, e.g., whether or not the language of a NLC grammar consists of non-planar graphs only. We will do this now.

It is a well-known fact (see, e.g., [1]) that a graph is not planar if and only if it has a subgraph that is homeomorphic to one of the Kuratowski graphs K_5 or K_6, where K_5 is the complete graph on 5 vertices and K_6 is the graph with 6 vertices shown in Fig. 11. This result will be used in the proof of our next theorem.

THEOREM 4. *It is undecidable whether or not the language of a given NLC grammar contains a planar graph.*

Proof. We will show that a decision procedure for this question leads to a decision procedure for the question of Theorem 3. To this aim, we provide a construction which, given an arbitrary RNLC grammar G, yields a RNLC grammar $\tilde{G}$ such that $L(\tilde{G})$ contains a planar graph if and only if $L(G)$ contains a totally disconnected graph. Let $G = (\Sigma, \Delta, P, Z)$. Then $\tilde{G} = (\tilde{\Sigma}. \tilde{\Delta}, \tilde{P}, \tilde{Z})$,

$$\tilde{\Sigma} = \Sigma \cup \{\not{C}\} \qquad \text{with} \quad \not{C} \notin \Sigma,$$

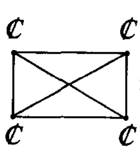
$$\tilde{\Delta} = \{\not{C}\},$$

$$\tilde{Z} = Z;$$

$\tilde{P}$ is constructed as follows.

(1) If $((l, D); C)$ is a production of P, then $((l, D); C \cup (\tilde{\Sigma} \times \{\not{C}\})) \in \tilde{P}$.

(2) For all $x \in \Delta$,

(x,  ; $\tilde{\Sigma} \times \tilde{\Sigma}$) $\in \tilde{P}$.

Obviously, $S(G) \subseteq S(\tilde{G})$. We now prove that indeed:

$L(G)$ contains a totally disconnected graph if ad only if $L(\tilde{G})$ contains a planar graph.

(a) "Only if": let H be a totally disconnected graph. It is obvious that by applying productions of type (2), a planar graph of $L(\tilde{G})$ can be derived from H.

(b) "If": let $\mathscr{D}$ be a derivation in $\tilde{G}$ from a planar graph. Then in $\mathscr{D}$, productions of type (2) can only be applied to isolated nodes. Indeed, if such a production was applied to a non-isolated node, this would result in a graph that contains as a subgraph a complete graph with eight nodes. (This follows from the fact that the connection relation of every production contains $\tilde{\Sigma} \times \{\mathcal{C}\}$, and from the definition of (2.) Now omit from $\mathscr{D}$ all steps in which a production of type (2) was used. The we get a derivation $\bar{\mathscr{D}}$ in G. Since the $\tilde{\Delta}$ consists of $\mathcal{C}$ only, the resulting graph $\bar{H}$ of $\bar{\mathscr{D}}$ has labels in Δ, and because the omitted steps are rewritings of isolated nodes, $\bar{H}$ is a totally disconnected graph of $L(G)$. ∎

We end this section by considering two decision problems the solutions of which follow from Theorem 3.

THEOREM 5. *It is undecidable whether or not, given an arbitrary NLC-grammar $G = (\Sigma, \Delta, P, C, Z)$, and two arbitrary subsets A, B of Δ, there is a graph H in $L(G)$ such that A is not adjacent to B in H.*

Proof. Let $A = \Delta$, $B = \Delta$, then $H \in L(G)$ and A is not adjacent to B in H if and only if H is totally disconnected. Hence the result follow from Theorem 3.

Remark. The theorem is also true if $A \cap B = \varnothing$, which is seen as follows. Let $\bar{G} = (\bar{\Sigma}, \bar{\Delta}, \bar{P}, \bar{Z})$ be a RNLC-grammar with $L(\bar{G}) = L(G)$. Let S, T be two new labels and let $A = \{S\}$, $B = \{T\}$. Replace $\bar{\Sigma}$ by $\{S, T\}$, add the set $\bar{\Sigma} \times \{S, T\}$ to the connection relation of every production and for every element X of $\bar{\Delta}$, add the production $((X, \overset{S}{\bullet}\ \overset{T}{\bullet});\ \{S, T\} \times (\bar{\Sigma} \cup \{S, T\}))$ to $\bar{P}$. Then clearly the new grammar derives a graph in which A is not adjacent to B if and only if $L(G) = L(\bar{G})$ contains a totally disconnected graph. (The "only if" part follows from the fact that a connection to a S- or T-labelled node can never be broken.) ∎

THEOREM 6. *Given an arbitrary NLC grammar G, it is undecidable whether or not $S(G)$ contains a graph H, such that the family $\{M \mid H \Rightarrow_G^* M\}$ is of bounded degree.*

Proof. By (17) from Section I it follows that we may suppose that G has no productions for elements of its terminal alphabet. Let $G = (\Sigma, \Delta, P, C, Z)$ and construct $\bar{G}$ from G as follows:

(1) add a new label, A say, to Σ and Δ $(A \notin \Sigma)$;

(2) add the set $\Sigma \times \{A\} \cup \{A\} \times \Sigma$ to C;

(3) add to P the productions

  (i) $(X, \overset{A}{\bullet})$ for every $X \in \Delta$,

  (ii) $(A, \overset{A}{\bullet}\ \overset{A}{\bullet})$;

then, for every $H \in S(G)$ holds that $\{M \mid H \Rightarrow^*_{\bar{G}} M\}$ is of bounded degree only if the set $\{K \in L(G) \mid H \Rightarrow^* K\}$ contains only totally disconnected graphs. Indeed, if for an arbitrary graph H over Σ we have $H \Rightarrow^*_G K$ then $H \Rightarrow^*_{\bar{G}} K$ and if $K \in L(G)$ and K is not totally disconnected, then we can derive from K a graph of arbitrary degree by the productions from (i) and (ii).

Now suppose n is the smallest positive integer such that there exists a $H \in S(\bar{G})$ for which $Z \Rightarrow^n_{\bar{G}} H$ and $\{M \mid H \Rightarrow^*_{\bar{G}} M\}$ is of bounded degree. We show that $H \in S(G)$. This is seen as follows:

—If $n = 0$, $H \in S(G)$ is obvious;

—If only production sof P are used in the derivation of H then obviously $H \in S(G)$.

—Otherwise, a production from $\bar{P} \backslash P$ must be applied to an isolated node. If omit this step from the derivation of H, then the obtained graph $\tilde{H}$ is such that $\{M \mid \tilde{H} \Rightarrow^*_{\bar{G}} M\}$ is of bounded degree (because only productions from $\bar{P} \backslash P$ can rewrite elements of Δ). This contradicts the minimality of n.

On the other hand, if $H \in L(G)$ and H is totally disconnected, $H \in S(\bar{G})$ and $\{M \mid H \Rightarrow^*_{\bar{G}} M\}$ is of bounded degree (The only production for elements of Δ in $\bar{G}$ are of type (i)). Hence we conclude that there exists a $H \in S(\bar{G})$ such that $\{M \mid H \Rightarrow^*_{\bar{G}} M\}$ is of bounded degree if and only if $L(G)$ contains a totally disconnected graph. Therefore, the desired result follows from Theorem 1. ∎

IV. The Second Basic Construction

In classical formal (string) language theory quite a number of decision problems were considered already. Hence a possible strategy to settle a decision problem concerning graph grammars is to "reduce" it to a decision problem in string grammars the solution of which is known. In this section we follow this strategy to settle solutions of a number of decision problems concerning NLC grammars. The result to which we reduce our problems is the undecidability of the emptiness problem for context sensitive (CS) grammars (see, e.g., [7]). Our "reduction technique" uses the following result from [6].

Theorem 7. *There exists an algorithm which, given an arbitrary CS string-grammar $\bar{G}$, constructs a NLC grammar G such that*

(1) *every element of $S_{\text{conn}}(G)$ is of the form*

$$\overset{₵}{\bullet}\!\!-\!\!\!-\!\!\overset{l_1}{\bullet}\!\!-\!\!\!-\!\!\overset{l_2}{\bullet}\!\!-\!\!\!-\!\!\overset{l_3}{\bullet}\!\!-\!\!\!-\!\!\overset{l_4}{\bullet}\!\!-\!\!\!-\cdots-\!\!\!-\!\!\overset{l_{n-1}}{\bullet}\!\!-\!\!\!-\!\!\overset{l_n}{\bullet},\ ₵ \notin \{l_1, l_2, \ldots, l_n\}.$$

(*₵ is a "special" symbol that cannot be rewritten.*)

(2) *If we define, for a graph H of the above form, $W(H) = l_1 l_2 \cdots l_n$, then* $\{W(H) \mid H \in L_{\text{conn}}(G)\} = L(\bar{G})$,

Since the emptiness problem for CS string-grammars is undecidable, the above $L(G)$ contains a connected graph.

It is undecidable whether or not, for an arbitrary NLC grammar G, $L(G)$ contains a connected graph.

THEOREM 9. *It is undecidable whether or not, for an arbitrary NLC grammar G, $L(G)$ contains a hamiltonian graph.*

A very natural problem to consider is whether or not the language of a NLC grammar is of bounded degree. We could not solve this problem, however, we can show that this problem is undecidable if one considers the connected language of a NLC grammar.

THEOREM 10. *Given an arbitrary NLC grammar G, it is undecidable whether or not $L_{\text{conn}}(G)$ is of bounded degree.*

Proof. We will show that we can construct a RNLC grammar $\bar{G}$ such that $L_{\text{conn}}(\bar{G})$ is of bounded degree if and only if $L_{\text{conn}}(G)$ is empty. Since, by (16) of Section I, we can construct a NLC grammar $\bar{\bar{G}}$ with $L(\bar{\bar{G}}) = L(\bar{G})$, the theorem then follows from Theorem 8.

From (17) of Section I we know that we can construct a RNLC grammar $\tilde{G} = (\tilde{\Sigma}, \tilde{\Delta}, \tilde{P}, \tilde{Z})$ such that $\tilde{P}$ contains no productions for lements of $\tilde{\Sigma}$ and $L(\tilde{G}) = L(G)$. Now let $\bar{G} = (\bar{\Sigma}, \bar{\Delta}, \bar{P}, \bar{Z})$ where

$$\bar{\Sigma} = \Sigma \cup \{\$\} \qquad \text{with} \quad \$ \notin \tilde{\Sigma},$$

$$\bar{\Delta} = \tilde{\Delta} \cup \{\$\},$$

$$\bar{Z} = Z,$$

$$\bar{P} = \tilde{P} \cup \left\{ \left(\left(X, \overset{X}{\bullet}\!\!-\!\!\!-\!\!\overset{\$}{\bullet} \right); \{X\} \times \bar{\Sigma} \right) \mid X \in \tilde{\Delta} \right\}.$$

Now if M is a graph, let $M_\$$ denote the full subgraph of M with label set $\{v \mid v \in V_M, \varphi_M(v) = \$\}$. Then for graphs H, $\bar{H}$ we have $H \Rightarrow_{\bar{G}} \bar{H}$ only if $H \backslash H_\$ \Rightarrow^*_{\tilde{G}} \bar{H} \backslash \bar{H}_\$$. (If the production, applied in $H \Rightarrow_{\bar{G}} \bar{H}$ belongs to $\bar{P} \backslash \tilde{P}$, then $H \backslash H_\$ = \bar{H} \backslash \bar{H}_\$$ and otherwise, this statement follows from the fact that the rewritten node does not belong to $H_\$$). From this we conclude that $H \in L(\bar{G})$ implies $H \backslash H_\$ \in L(\tilde{G})$. We now prove that for $H \in S(\bar{G})$, H is connected if and only if $H \backslash H_\$$ is connected. This follows from the fact that every \$-labelled node of H has exactly one neighbor, and this neighbor has a label in $\tilde{\Delta}$. We show this by induction on the length of a derivation for H.

(i) If $H = \bar{Z}$, then the result is obvious (H has no \$-labelled nodes).

(ii) Assume that the result holds for every graph H that can be derived from $\bar{Z}$ in

$\bar{G}$ in at most $n-1$ steps. Let M be such that $\bar{Z} \Rightarrow_{\bar{G}}^{n-1} H \Rightarrow_{\bar{G}} M$. By the induction hypothesis, H satisfies the condition. For the nth step of the derivation of M we have two possibilities: either a production of $\bar{P} \backslash \tilde{P}$ is applied and then M satisfies the condition, or a production of $\tilde{P}$ is applied and then, since in $\tilde{P}$ no element of $\tilde{\Delta}$ can be rewritten and because of the induction hypothesis, the rewritten node of H has no \$-labelled neighbors. Therefore M satisfies the condition also in this case.

Now if $L_{\text{conn}}(\bar{G})$ is not of bounded degree, it is nonempty and it contains a graph H with $H \neq H_{\$}$. We conclude that $H \backslash H_{\$}$ is an element of $L_{\text{conn}}(\tilde{G}) = L_{\text{conn}}(G)$ and thus, $L_{\text{conn}}(G)$ is nonempty. On the other hand, if $H \in L_{\text{conn}}(G)$ then $H \in S_{\text{conn}}(\bar{G})$ and by applying productions from $\bar{P} \backslash \tilde{P}$ we clearly can derive from H connected graphs of an arbitrary degree. ∎

We also have the following reduction of the problem of bounded degreeness for NLC grammars.

THEOREM 11. *For an arbitrary NLC grammar G, it is decidable whether or not $L(G)$ is of bounded degree if and only if it is decidable whether or not $L(C) \backslash L_{\text{conn}}(G)$ is of bounded degree.*

Proof. For any NLC grammar $G = (\Sigma, \Delta, P, C, Z)$ we can construct a NLC grammar $\bar{G}$ such that (1) every element of $L(\bar{G})$ is disconnected, and (2) $L(\bar{G})$ is of bounded degree if and only if $L(G)$ is of bounded degree. The construction is easy: add a new label, L, say, to Σ and Δ, and add an isolated, L-labelled node to Z. Then every element of $S(\bar{G})$ consist of two parts: an isolated, L-labelled node and an element of $S(G)$. ∎

The problem whether or not the languages generated by two (graph) grammars intersect is a quite standard problem to consider. Its solution for NLC grammars is provided now.

THEOREM 12. *Given two NLC-grammars G_1 and G_2, it is undecidable whether $L(G_1) \cap L(G_2) = \varnothing$.*

Proof. Let $\bar{G}_1 = (\bar{\Sigma}_1, \bar{\Delta}_1, \bar{P}_1, \bar{Z}_1)$ be an arbitrary CS grammar. Let $G_1 = (\Sigma_1, \Delta_1, P_1, C_1, Z_1)$ be the corresponding NLC grammar from Theorem 7. Let G_2 be the NLC grammar $G_2 = (\Sigma_2, \Delta_2, P_2, C_2, Z_2)$, where

$$\Sigma_2 = \Sigma_1 \text{ (hence } \text{¢} \in \Sigma_2),$$

$$\Delta_2 = \Delta_1 \text{ (hence } \text{¢} \in \Delta_2),$$

$$Z_2 = \overset{\text{¢}}{\bullet},$$

$$F_2 = \left\{ \left(\text{¢}, \overset{\text{¢}}{\bullet} \!\!-\!\!-\!\! \overset{X}{\bullet} \right) \,\middle|\, X \in \Delta_1 \right\}$$

$$C_2 = \Delta_1 \backslash \{\text{¢}\} \times \Delta_1 \backslash \{\text{¢}\}.$$

Then $L(G_2)$ is the set of chains with labels in Δ_1. We clearly have $L(G_1) \cap L(G_2) = L_{\text{conn}}(G_1)$ and therefore, the result follows from Theorem 8. ∎

We now give a technical lemma that is a variation of Theorem 7 and that will be useful in the proof of our next three undecidability results, which conclude this section.

LEMMA 1. *There exists an algorithm which, given an arbitrary CS string-grammar $\bar{G}$, constructs a NLC grammar $\tilde{G}$ such that*

(1) *every element of $S_{\text{conn}}(\tilde{G})$ is of the form*

$$\overset{\not{C}}{\bullet}\!\!-\!\!\!-\!\!\overset{l_1}{\bullet}\!\!-\!\!\!-\!\!\overset{l_2}{\bullet}\!\!-\!\!\!-\!\!\overset{l_3}{\bullet}\!\!-\!\!\!-\!\!\overset{l_4}{\bullet}\cdots\!\!-\!\!\!-\!\!\overset{l_n}{\bullet}\!\!-\!\!\!-\!\!\overset{\$}{\bullet},$$

$\not{C}, \$ \notin \{l_1, l_2, \ldots, l_n\}$ ($\not{C}$ and $\$$ are "special" symbols that cannot be rewritten).

(2) *If we define, for a graph H of the above form, $U(H) = l_1 l_2 \cdots l_n$, then $\{U(H) \mid H \in L_{\text{conn}}(\tilde{G})\} = L(\bar{G})$,*

Proof. By (16) of Section I, it suffices to provide a construction of a RNLC grammar $\hat{G}$ with the above properties. Such a construction can be obtained by the following changes in the construction of the proof of Theorem 7 (Theorem 9 in [6]).

(a) Add the new symbol $\not{C}$ to Σ and Δ.

(b) If $\bar{Z} = l^{(1)}$ then let

$$Z = \overset{\not{C}}{\bullet}\!\!-\!\!\!-\!\!\overset{l_1^{(1)}}{\bullet}\!\!-\!\!\!-\!\!\overset{\$}{\bullet}.$$

(c) If R is a production of G with only one node in its right-hand side, then add $\Sigma \times \{\$\}$to $E(R)$.

(d) If R is a production of G of the form

$$\left(\left([ABC]_i, \overset{w_i}{\bullet}\!\!-\!\!\!-\!\!\overset{X_{\sigma(i)}}{\bullet}\!\!-\!\!\!-\!\!\overset{Y_{\sigma(i)}}{\bullet}\!\!-\!\!\!-\!\!\overset{[Z]_i}{\bullet}\right); E(R)\right)$$

then add $\{([Z]_i, \$)\}$ to $E(R)$. ∎

THEOREM 13. *Given an arbitrary NLC grammar $G = (\Sigma, \Delta, P, C, Z)$, and two arbitrary subsets A, B of Δ, it is undecidable whether or not A and B are connected in G.*

Proof. Let $\bar{G}$ be an arbitrary CS string-grammar. Let $\tilde{G}$ be as in the statement of Lemma 1. Let $A = \{\not{C}\}$, $B = \{\$\}$. The clearly A is connected to B in $\tilde{G}$ if and only if $L(\bar{G})$ is non-empty. The desired result now follows from the fact that the emptiness problem for CS string-grammars is undecidable. ∎

THEOREM 14. *Given an arbitrary NLC grammar G, it is undecidable whether or not $L(G)$ contains a graph that is not* 1*-connected.*

Proof. Let $\bar{G}$ be any CS string-grammar. Let $\tilde{G}$ be as in the statement of Lemma 1. Let $\tilde{G}_1$ result from $\tilde{G}$ by adding to the axiom $\tilde{Z}$ of $\tilde{G}$ an edge incident to the ¢-labelled node and to the \$-labelled node of $\tilde{Z}$. Then clearly $L(\tilde{G}_1)$ contains a graph that is not 1-connected if and only if $L(\bar{G})$ is nonempty. We then use again the undecidability of the empthiness problem for CS string-grammars to obtain the desired result. ∎

THEOREM 15. *Given an arbitrary ANLC grammar G, it is undecidable whether or not $L(G)$ is of bounded degree.*

Proof. Let $\bar{G}$ be an arbitrary CS string-grammar, and let $\tilde{G} = (\tilde{\Sigma}, \tilde{\Delta}, \tilde{P}, \tilde{C}, \tilde{Z})$ be the NLC grammar from the statement of Lemma 1. Let $\tilde{G}_1 = (\tilde{\Sigma}_1, \tilde{\Delta}_1, \tilde{P}_1, \tilde{C}_1, \tilde{Z}_1)$, where

$$\tilde{\Sigma}_1 = \tilde{\Sigma},$$
$$\tilde{\Delta}_1 = \tilde{\Delta},$$
$$\tilde{Z}_1 = \tilde{Z},$$

$$\tilde{P}_1 = \tilde{P}_1 \cup \Pi_1 \cup \left\{ \left(\$, \overset{\$}{\bullet} \;\; \overset{\$}{\bullet}\right) \right\} \qquad \text{with} \quad \Pi_1 = \{(X, \Lambda) \mid X \in \tilde{\Delta}\}, \text{and } \tilde{C}_1 = \tilde{C} \cup \{(\$, \text{¢})\}.$$

We will show that $L(\tilde{G}_1)$ is of bounded degree if and only if $L(\bar{G})$ is empty. The theorem then again follows from the undecidability of the emptiness problem for CS string-grammars.

First we show that there exists a H in $L(\tilde{G}_1)$ of the form

$$\text{¢} \quad X_1 \quad X_2 \quad \cdots \quad X_n \quad \$$$

if and only if there exist $\alpha_1, \alpha_2, \ldots, \alpha_{n+1}$ in $(\tilde{\Delta}_1)^*$ such that $\alpha_1 X_1 \alpha_2 X_2 \alpha_3 \cdots \alpha_n X_n \alpha_{n+1} \in L(\bar{G})$.

The if part follows from Lemma 1 and the form of the productions from Π_1. We prove the only if part by induction on the length of a derivation of H.

(1) If $H = \tilde{Z}_1$ then the result trivially holds.

(2) Let us assume that the results holds if $Z_1 \Rightarrow^{n-1}_{\tilde{G}_1} H$ and that $H \Rightarrow_{G_1} \hat{H}$. If no erasing production is used in deriving $\hat{H}$ from $\tilde{Z}_1$ then the result follows directly from Lemma 1. If a production $A \to \Lambda$ is used, it follows from the construction from the proof of Theorem 7 (see [6]) that by omitting this production we get a connected graph, $\bar{H}$, of the form

$$\text{¢} \quad X_1 \quad X_2 \quad X_3 \quad \cdots \quad X_{i-1} \quad A \quad X_i \quad \cdots \quad X_n \quad \$$$

($\bar{H}$ differs from $\hat{H}$ only by the extra A-labelled node). We have $Z_1 \Rightarrow_{G_1}^{n-1} \bar{H}$ and we can use the induction hypothesis: For some

$$\alpha_1, \alpha_2, \ldots, \alpha_{i-1}, \bar{\alpha}, \bar{\bar{\alpha}}, \alpha_{i+1}, \ldots, \alpha_{n+1},$$

$$\alpha_1 X_1 \alpha_2 X_2 \cdots \alpha_{i-1} X_{i-1} \bar{\alpha} A \bar{\bar{\alpha}} X_i \alpha_{i+1} \cdots \alpha_n X_n \alpha_{n+1} \in L(\bar{G}) \text{ and we let}$$

$$\alpha_i = \bar{\alpha} A \bar{\bar{\alpha}}.$$

Now suppose $L(\bar{G})$ is empty. Then no graph of the form $₵ \text{---} \$$ occurs in $L(\tilde{G}_1)$. Since ₵ and \$ cannot be rewritten in $\tilde{G}$ (see the proof of Theorem 3 in [6]), it follows that ₵ and \$ are not adjacent in $\tilde{G}_1$. We conclude that $L(\tilde{G}_1)$ is of bounded degree.

To prove that $L(\tilde{G}_1)$ is of bounded degree only if $L(\tilde{G})$ is empty, suppose $L(\bar{G})$ is not empty. Then $L(\tilde{G}_1)$ contains as graph of the form $₵ \text{---} \$$. Clearly, by successive applications of the production $\$ \rightarrow \$\,\$$ we can derive graphs of arbitrary degree from this graph. ∎

References

1. S. S. Anderson, "Graph Theory and Finite Combinatorics," Markham, Chicago, 1970.
2. V. Claus, H. Ehrig, and G. Rozenberg (Eds.), "Graph Grammars," Lecture Notes in Computer Science No. 73, Springer-Verlag, Berlin/Heidelberg/New York, 1980.
3. M. Nagl, A tutorial and bibliographical survey of graph grammars, *in* "Graph Grammars" (V. Claus *et al.*, Eds.), Lecture Notes in Computer Science No. 73, Springer-Verlag, Berlin/New York/Heidelberg, 1980.
4. H. Ehrig, Introduction to the algebraic theory of graph grammars, *in* "Graph Grammars" (V. Claus *et al.*, Eds.), Lecture Notes in Computer Science No. 73, Springer-Verlag, Berlin/New York/Heidelberg, 1980.
5. D. Janssens and G. Rozenberg, On the structure of node-labelled graph languages, *Inform. Sci.* **20** (1980), 191–216.
6. D. Janssens and G. Rozenberg, Restrictions, extensions and variations of *NLC* grammars, *Inform. Sci.* **20** (1980), 217–244.
7. A. Salomaa, "Formal Languages," Academic Press, New York/London, 1973.